AD-A048 292 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/4
NEW MULTIPLEXING AND DATA FLOW CONTROL TECHNIQUES FOR TIME SHAR--ETC(U)
JUN 77 G C LOWE

UNCLASSIFIED RAE-TR-77085 DRIC-BR-59171 NL

1 OF 1
AD
A048292

END
DATE
FILMED
2-78
DDC

UNLIMITED (14) RAE -TR -77Ø85

(18) DRIC (19) BR59171

1

ROYAL AIRCRAFT ESTABLISHMENT

*

(9) Technical Rep[redacted]t. 77085

(11) Jun[redacted]77

(12) 41p.

(6)

# NEW MULTIPLEXING AND DATA FLOW CONTROL TECHNIQUES FOR TIME SHARED DATA ACQUISITION SYSTEMS.

by 310 450

(10) G.C./Lowe

*

Procurement Executive, Ministry of Defence
Farnborough, Hants

# REPORT DOCUMENTATION PAGE

Overall security classification of this page

**UNLIMITED**

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

| | | | |
|---|---|---|---|
| 1. DRIC Reference (to be added by DRIC) | 2. Originator's Reference<br>RAE TR 77085 | 3. Agency Reference<br>N/A | 4. Report Security Classification/Marking<br>UNLIMITED |

| | |
|---|---|
| 5. DRIC Code for Originator<br>850100 | 6. Originator (Corporate Author) Name and Location<br>Royal Aircraft Establishment, Farnborough, Hants, UK |
| 5a. Sponsoring Agency's Code<br>N/A | 6a. Sponsoring Agency (Contract Authority) Name and Location<br>N/A |

7. Title
New multiplexing and data flow control techniques for time shared data acquisition systems

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

| | | | | | |
|---|---|---|---|---|---|
| 8. Author 1. Surname, Initials<br>Lowe, G.C. | 9a. Author 2 | 9b. Authors 3, 4 .... | 10. Date<br>June 1977 | Pages<br>39 | Refs.<br>8 |
| 11. Contract Number<br>N/A | 12. Period<br>N/A | 13. Project | 14. Other Reference Nos.<br>IT 154 | | |

15. Distribution statement
(a) Controlled by – Unlimited
(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Time division multiplexing*. Decommutators*. Data acquisition*.
Data processing*.

17. Abstract

The Report describes a novel and versatile method for the control of the order of sampling data sources in flight test data acquisition systems, which was needed, because existing systems were either bulky or lacked the necessary flexibility and did not provide a wide range of sampling rates. A new method for dealing with the synchronising and housekeeping information that is inherent in these systems is also described.

F5910/1

UDC 681.325 : 681.3.025

## ROYAL AIRCRAFT ESTABLISHMENT

Technical Report 77085

Received for printing 20 June 1977

# NEW MULTIPLEXING AND DATA FLOW CONTROL TECHNIQUES FOR TIME SHARED DATA ACQUISITION SYSTEMS

by

G. C. Lowe

## SUMMARY

The Report describes a novel and versatile method for the control of the order of sampling data sources in flight test data acquisition systems, which was needed because existing systems were either bulky or lacked the necessary flexibility and did not provide a wide range of sampling rates. A new method for dealing with the synchronising and housekeeping information that is inherent in these systems is also described.

Departmental Reference: IT 154

## LIST OF CONTENTS

## 1 INTRODUCTION

In a data acquisition system it is normal practice for a number of data sources to share part of the system, in particular the data recorder or communication link. This sharing is most commonly based on time in which case each source is connected to the system in turn for a fixed time interval. Some means of controlling the order in which the sources are connected must therefore be provided and if the system is to be used for a variety of tasks the controlling mechanism should provide the user with a means of specifying the data sources and corresponding sampling rates that are required on a particular occasion. During the development of a general purpose flight test data acquisition system[1,2] it was found that most existing methods were either very bulky or did not provide adequate flexibility and that no method existed that could provide a wide range of sampling rates. A new method[3,4] was therefore developed to give great versatility in programming at small cost in airborne equipment complication and is described in this Report.

An associated computer-based data replay system was developed at the same time as the airborne equipment. Efforts were made to achieve a balance between the specialised requirements of the acquisition and processing parts of the overall system. This system when constructed worked well but had some operational weaknesses. A later decision to increase the data capacity of the system provided an opportunity to introduce a powerful editing facility which considerably improved the overall system performance in several respects.

The new techniques used are described in the present Report, the main aims of the work being to develop an easily used programming system requiring the minimum of hardware and yet imposing the minimum of restrictions on the user. It was required that the necessary 'housekeeping' information (synchronisation, timing etc) should be inserted automatically during recording and that editing or selection of required data on replay should be simple and rapid.

## 2 MULTIPLEXING METHOD

### 2.1 Review of the problem

Early time division multiplexed systems used motor driven mechanical
switches[5] to connect analogue sources in turn to an analogue communications
85 link. The basic sampling rate was defined by the speed of rotation of the
switch rotor, higher rates being obtained by connecting several equally spaced
contacts together and lower rates by connecting one or more contacts to the
wipers of secondary switches driven from the same motor via a reduction gearbox.

A number of technical terms (*eg* sub-commutation, super commutation, field, frame, sub-frame, major frame etc) have been devised to describe various properties of the sampling patterns produced by these motor driven devices. In general the use of these terms leads to confusion, due to varying definitions, and only the terms field (the data generated during one complete cycle of the sampling process) and pattern (a particular sequence of channel numbers) will be used in the following discussion.

Individual switch elements[6] of a variety of types (*eg* electromechanical relay, reed relay, transistor, double emitter transistor, junction FET etc) have been used to replace rotary mechanical switches but the most successful to-date is the CMOS field effect device. The individual electronic elements were first used together with ring-counters to provide a direct analogue of the motor driven switch and later with read-only (*eg* diode matrix) or random access memories (*eg* core store) to give more versatile systems. In the latter case the store locations are examined sequentially, their contents defining which switch should close, thus allowing any switching sequence to be specified but with a field length limited to the number of words stored.

Digital control and encoding techniques are being increasingly used in data acquisition systems in association with the above developments to handle both analogue and digital sources and to route the data in digital form either to a digital tape recorder or transmission link. The expanding size and versatility of these systems led to the need for an improved means of selecting the sampling sequence and this is the subject of the present work.

The aim of this work was to develop a multiplex switch control system which was versatile and yet easy to use. The acquisition equipment was required to be as small as practical and to make efficient use of the recorder or transmission link. For a system to be efficient two criteria must be satisfied; sampling rates close to the values desired by the user must be available (to ensure that unnecessary over-sampling does not occur) and the samples must occupy the bulk of the data field, with a minimum of padding to make the sampling pattern a practicable one. Both of these criteria are influenced by the choice of the relative sampling rates of the different channels.

For the purposes of this Report a padding slot is defined to be a position in the data field occupied by padding information, and the relative sampling rates to be the ratios of the absolute values of the sampling rates to the lowest absolute rate in use, scaled by the smallest constant such that all the 085 relative rates available are integers. Using these definitions, it can be demonstrated that, in any system in which it is desired to sample individual inputs at equally spaced instants in time, the maximum number of sampling rates, within a given range, and the lowest number of padding slots required occurs when all the sampling rates are related to each other by powers of 2. In this case if a system has K relative rates the kth relative rate $r_k$ is given by:-

$$r_k = 2^{k-1} \tag{1}$$

for

$$1 \leqslant k \leqslant K .$$

If the number of sources sampled at rate $r_k$ is $N_k$ then the field length F is given by:-

$$F = \sum_{k=1}^{K} N_k 2^{(k-1)} + P \tag{2}$$

where P is the number of padding slots that must be added to produce a viable field, as discussed above, and lies in the range $0 \leqslant P \leqslant 2^{(K-1)} - 1$ such that

$$F = 2^{K-1} E_K \tag{3}$$

where $E_K$ is the repetition interval of sources sampled at the highest relative rate. It can be seen from equation (2) that the value of F increases rapidly as K and N (the total number of sources) increase and that it soon becomes impracticable to store the whole field pattern in a sequentially addressed memory. The storage capacity required may be reduced if a suitable logic/arithmetic unit is used to compute the successive elements of the pattern. In this case only those constants required for the calculation need be stored. The minimum storage capacity, consistent with operational flexibility, is discussed below followed by a description of one possible pattern generation process.

## 2.2 Storage requirements

### 2.2.1 General considerations

Consider the case of a system of the form shown in Fig 1. The switch controller causes the switches SW1 to SWN to close and open in the desired sequence. Its output may be considered to be a sequence of numbers corresponding to the addresses of the switches to be closed. The minimum information (in

085

terms of the number of words to be stored), required by the controller to enable it to function satisfactorily may be deduced as follows.

Assume initially that the sources S1 to SN are numbered in order of sampling rate required, the highest rate applying to source S1. Since there are only a finite number of sampling rates available this ordering will result in the grouping together of the sources to be sampled at a given rate which will therefore be identified by a consecutive block of numbers. It is possible to generate sampling patterns for such a system from a knowledge of only one parameter per sampling rate available (*ie* for a system having K relative rates only K words are required). The number of sources to be sampled affects the size of the word stored but not the number of words. The parameter involved may be defined in a number of ways, for example it can be defined as the number of sources to be sampled at a given rate or it could be the last (*ie* highest) address for each block. In practice the definition that simplifies the physical implementation would be chosen. The store holding these parameters will be called the Scan Store and is discussed in detail below.

In many applications it is impossible, or at least undesirable, to physically connect specific sources to specific switches. This is true, for example, in the case of large aircraft where the data acquisition system may consist of several acquisition units strategically sited throughout the aircraft each unit being connected (probably via a digital link) to a central processing and record unit. In this case and in the majority of cases where ease of change is required it is desirable that a means of re-allocating switch addresses should be provided. This is easily accomplished, through the use of a second store (the first being the Scan Store), in the following manner. The sources are connected to the most convenient, probably the nearest, switches. Under these conditions there is no relationship between the switch addresses and the sampling rates required. The desired order can be restored (*ie* the system can be effectively rewired) if the switch addresses, associated with sources to be sampled at the same rate, are held in consecutive locations in the second store which is called the Translator. This is illustrated in Fig 2b. Each block of entries contains the switch addresses of sources to be sampled at a given rate. The output of the switch controller of Fig 1 generates the appropriate Translator Store addresses causing the desired sequence of numbers (switch addresses) to appear at the output of the store. For example if the switch controller generates the consecutive set of addresses $b_0$ to $b_2$ the multiplexing switches will close in the order SW5, SW2 and SW7. 085

A sampling pattern may be defined (as explained above) using a minimum of one parameter per sampling rate available. This may be achieved by arranging for the blocks of entries (A, B and D of Fig 2b) for each rate to be consecutive and in sampling rate order. This approach does not make the best use of the Translator capacity and operational flexibility and efficiency can be improved if two parameters per rate are used. Each of the blocks of entries may then be stored anywhere within the Translator. The position and size of the blocks are a possible pair of parameters which may be defined by the start and stop numbers (addresses of the first and last entries of the block) which are held in the Scan Store (*eg* $b_0$ and $b_2$ of Fig 2a). The order of the entires in the Scan Store indicates the rate at which the items in each block are to be sampled. An unused sampling rate may be indicated by a dedicated number denoted by X in Fig 2a.

The sampling program in use may be changed by changing the Scan Store contents. Different blocks, or parts of blocks, in the Translator may be used in different programs and this could be accomplished by storing the information on each sampling program in different areas within the Scan Store. The scan could then be changed during the data acquisition process (*eg* between take off and cruise) thus providing considerable operational flexibility.

### 2.2.2 Physical implementation

The discussion above indicates that the total storage capacity needed for a given application depends on the flexibility required. In the case of a simple system, where the sources are connected in sampling rate order, no Translator is required and a Scan Store containing a minimum of K words (*ie* one word per sampling rate available) is needed. However most systems require a degree of flexibility and this may be provided by a Scan Store containing 2K + 2 (two per rate plus two spare, see below) words per program and a Translator containing at least N words (one per source). Thus a system having 15 binary related sampling rates (a range of 16384:1) and 1024 sources would require only 1056 words of storage and yet be capable of generating sampling patterns of up to 16777216 words. In a practical system other factors such as synchronisation and demultiplexing requirements normally impose an upper limit on field length if system word lengths are to be kept within bounds.

The capacity defined above allows for two extra words in the Scan Store, the repetition interval ($E_K$) and the system sampling rate, which are used to simplify and control the pattern generation process as discussed below.

## 2.3 Generation of address sequence

In most time division multiplex data acquisition systems a pattern of switch addresses must be produced which causes samples to be taken from specific sources at rates requested by the user. The versatility and ease of use of a system depend on the technique used to generate this address sequence.

In the case of the sequentially addressed memory the whole of the pattern must be predetermined. Since the pattern may be tens of thousands of words long some assistance, probably involving the use of a general purpose computer, with its derivation from the user requirement, is normally needed. As stated earlier any pattern may be generated provided the store capacity is adequate so that it is a reasonably simple matter to check whether a given requirement can be accommodated.

One method, that allows a reduction in storage capacity, is to base the system on a computer (software) analogue of the earlier mechanical switch arrangements (see section 2.1). A system of this type would implement some of the standard computer instructions (such as skip and jump) and use these in place of the wire links of the earlier system. Obviously a wide range of solutions are possible with this approach.

The degree of sophistication depends, to some extent, on the complexity of the processing unit used. A fixed number, probably small, of basic sampling rates would be provided corresponding to the switch sections coupled through gearboxes in the mechanical analogue. The interswitch wiring of the mechanical arrangement would be replaced by computer type instructions (*eg* jump, skip etc) which would cause the system to perform a search for the next channel address. All signal sources, other than those that are sampled at the system's basic rate, require more than one entry in the systems store. For example repeated, equally spaced entries of the source address are needed if a higher rate is required or alternatively some form of routing information is required to refer the search to another part of the program when a lower rate is required. The number of extra entries, in addition to the minimum of one per source, that are required varies widely and depends largely on the sampling requirements of the user. The conversion of the user's requirement into a viable list of instructions is a complex process and would probably require the assistance of a general purpose computer. It is not until a satisfactory list of instructions is compiled that a decision can be made as to whether or not the task is within the capabilities of the equipment (as regards storage capacity etc).

To overcome the disadvantages of such systems a method has been developed which makes use of the principles of the Scan and Translator Stores of section 2.2. Fig 3 illustrates how, for the simple 7 channel 4 sampling rate example given in Fig 2, the Translator addresses ($a_0$, $b_1$, $b_2$ etc) are fitted together to give the final sampling pattern. Priority is given to those addresses corresponding to sources that are to be sampled at the highest rate (block A in Fig 2b) and time slots remaining unused are filled with addresses corresponding to progressively lower sampling rates (from blocks B and D of Fig 2b). This process is carried out automatically by a sampling controller (see section 2.4) which uses information, supplied by the user, that is held in the Scan Store.

The repetition interval ($E_K$) of entries at the highest rate must be evaluated before the pattern of Fig 3 can be constructed. The value of $E_K$, which is defined by equations (1) and (2), may be obtained using the table shown to the left of Fig 3. The sum of the products of the relative sampling rates and corresponding number of sources, 23 here, must be rounded up to the nearest integral multiple of the highest relative rate, thus ensuring that samples are taken, from a given source, at equally spaced instants in time. In this example one padding slot, labelled P in Fig 3, is required to increase the sum from 23 to 24 which is a multiple of 8. The repetition interval ($E_K$) is therefore 3 and the address $a_0$ repeats every third time slot. The repetition interval for entries at the second highest relative rate is doubled and is 6 in this example. Similarly the interval doubles for each progressively lower rate.

It is convenient, in practice, to divide the pattern into sections, each section at a given rate being one repetition interval long (see Fig 3). A signal is generated at the end of each of these repetition intervals to indicate when the sources to be sampled at that rate become eligible for another round of sampling. It is termed the Rate Reset signal. Once sampled a given source becomes ineligible until the next relevant Rate Reset signal occurs. The eligible sources are sampled in order of their priority. Priority is allocated on a basis of sampling rate required with the highest priority being given to the highest rate. Sources that share a given rate are taken in the order in which their addresses appear in the Translator.

## 2.4 Physical realization

### 2.4.1 General considerations

There are several approaches to the problem of designing a sampling controller based on the technique described in section 2.3. Some of the factors affecting the choice of approach are:-

the speed of operation required,
the number of relative sampling rates desired,
the type and arrangement of stores used, and
the need, if any, for compatibility with existing equipment, for example data links and data reduction systems including demultiplexing equipment.

The usable operating speed, power consumption and cost of storage devices usually increase together. It is therefore desirable to reduce the extent of manipulation needed during one sample period to a minimum and to ensure that it is independent of the number of relative sampling rates provided or in use at any time. If these requirements are to be met the processing logic must be informed of those sampling rates which are not being used otherwise it would need to search several, and in the limit all of the entries in one word period in order to find one in use. The principles of one particular solution are discussed below as a means of illustrating what the implementation of this technique involves.

### 2.4.2 System description

A block diagram of the system is given in Fig 4a which could be constructed using standard integrated circuit logic elements such as transistor-transistor logic. It consists of the Scan and Translator Stores which hold the information, supplied by the user, that defines the sampling pattern to be generated (see section 2.2) and a Working Store which holds information, one number for each sampling rate, that defines which translator address is due to be issued next at each sampling rate. The numbers held in the Working Store lie in the range defined by the corresponding stop and start numbers in the Scan Store, for example, referring to Fig 3, the entries at the end of time slot 8 would be $a_0$, $b_2$, X (not used) and $d_1$ .

The Rate-Resets Generator, which consists of a programmable divider ($\div E_K$) followed by a cascaded chain of binary dividers, generates the rate reset signals described in section 2.3 and indicated on Fig 3. These reset signals, together with other control information, are used by the Rate Selector to determine the sampling rate of the next source to be sampled. The Rate

Selector consists of a Rates-In-Use Mask which comprises a register and a set of gates, a Rates-Pending Register and a Priority Encoder. Access to each element of the registers is possible and is controlled either by the System Timing and Control or the Decision Logic.

The Decision Logic contains a comparator which compares the Working Store contents with the stop number from the Scan Store and an adder which is used to increment the contents of the Working Store.

The method of operation of each of these parts of the system is described in the following section.

### 2.4.3 System operation

The system must be initialised before it will generate the desired sampling pattern. This involves establishing which rates are to be used, and the value of $E_K$ required, for the particular scan selected. In order to achieve this the contents of the Scan Store are examined on a cyclic basis, one word being looked at during the last third of each sample period (see Fig 4b). The Scan Store address used is incremented by one each time, so that the entire contents are eventually examined, over a period termed a slow cycle. During this cycle the value of $E_K$ is extracted and entered into the Rate-Resets Generator and also any unused rates (indicated by start and stop addresses of all ones, the unprogrammed state) are detected and the appropriate entry made in the Rates-In-Use Mask.

Once a slow cycle of the Scan Store has been completed all the essential information will have been established and the sampling controller will produce the required sampling pattern. The process by which this is achieved will now be described in outline.

As will be seen below, towards the end of any sample period the Working Store contains information that defines which translator address should be issued next for each sampling rate in use and the Rates-Pending Register indicates at which rates there are sources that are eligible for sampling (see section 2.3). The Priority Encoder generates a number which indicates the most significant position that is set in the Rates-Pending Register and therefore indicates the highest rate which is eligible for use. This number is used to address the Working Store causing the next translator address at this rate to appear at the output of the store at the end of the sample period. 085

At the start of the next sample period the output of the Working Store is transferred to the output register and, after translation, to the multiplexing switches. The contents of the Working Store and the Rates-Pending Register are then updated, to take account of the sample request just issued. This occupies the first two thirds of the sample period (see Fig 4b). During the first third of the period the Working Store output is compared in the Decision Logic with the stop number from the Scan Store. If it is less than the stop number one is added and the new value returned to the Working Store. If it is equal to or greater than (a fault condition) the stop number there are no more sources remaining to be sampled at the rate in question and the appropriate bit in the Rates-Pending Register is cleared and the start number is written into the Working Store during the second third of the sample period. If there are no bits set in the Rates-Pending Register the sample period is a padding slot and the Priority Encoder causes the Decision Logic to issue a specific address, to the Translator indicating that padding is to be inserted. The number of times the contents of the Scan Store are examined during a sample period is small and independent of the number of sampling rates available. The system is also informed of the rates in use, thus satisfying the requirements, discussed in section 2.4.1, for optimum performance.

### 2.4.4 Methods of increasing versatility

It is probable that data acquisition tasks will arise which do not require the full range of relative sampling rates provided. To avoid artificially long fields or excessive padding or both in these cases, some, preferably automatic, means of adjusting the system must be provided. Since the position of an entry in the Scan Store indicates the rate at which the associated sources are to be sampled (see section 2.2.1) the solution to this problem is simplified if the user always fills the Scan Store from one end. If, for example, the lowest position is always used then there will be a consecutive block of unused rates at the top of the Scan Store when the full sampling rate range is not employed. The contents of the Rates-In-Use-Mask (see Fig 4a) could be examined to determine the highest rate in use. This information could then be used to artificially shorten the dividing chain of the Rate-Resets-Generator by causing the output of the $E_K$ divider to be connected to the appropriate tap down the chain (this is not shown in Fig 4a) thus achieving the desired system modification. An alternative would be to use a similar scheme but always to use the highest rate instead of the lowest. In this case the pointer would indicate the point along the counter chain from which the 'pattern complete flag' (also the lowest rate reset or overall reset) should be derived. This approach is probably the simplest to implement but is a little less straightforward for the user since the position of the lowest relative rate (*ie* equal to once per pattern) in the table (see section 2.5) varies according to the relative rate range employed.

It is probable that the divider chain, in new and unrestricted designs, would be binary for the reasons discussed earlier (see section 2.1). However if the system must be compatible with existing data processing systems, especially demultiplexers, a mixture of division ratios could be provided. For example ratios of 8, 3, 2, 4, 3, 2 and 2 or 8, 3, 2, 4, 2, 3 and 2 will result in patterns which are compatible with 4650 - Variant II telemetry equipment[7] provided the appropriate number of channels are sampled at each rate though in this case other measures must be adopted to ensure that the synchronising information is correctly inserted. It would be possible to arrange for the division ratios to be programmable in the same way as the first stage (÷ $E_K$, Fig 4a) but the additional complexity involved is unlikely to be justifiable in mobile applications such as aircraft and missiles.

## 2.5 Program writing

The conversion of the information that defines the sampling requirement into a form that is suitable for entry into the Scan Store and Translator is a straightforward process that does not require the user to have a detailed knowledge of the system. The actual steps involved in this process are dependent on the specific physical implementation used and on the requirements of the rest of the system (synchronisation and other housekeeping functions). A typical procedure would be to:-

(a) List the sources to be sampled in order of their bandwidth or desired sampling rate, including housekeeping items where appropriate.

(b) Partition the list into suitable groups and allocate one of the available relative rates to each group.

(c) Calculate the pattern length and hence the repetition interval of sources sampled at the highest relative rate in use ($E_K$) (see below).

(d) Choose a system sampling rate that yields an acceptable absolute value for the lowest relative rate.

(e) If a tape recorder is to be used choose a tape speed.

(f) Allocate blocks of storage in the Translator for each of the groups of sources defined in step (b).

(g) Enter the START and STOP numbers of each group together with the master reset number, system sampling rate and tape speed.

(h) Enter the source addresses in the selected locations in the Translator.

The first three steps represent a method of defining the requirement which must be carried out in one form or another regardless of the multiplexing method employed. This is true also of the next two steps in the case of systems offering these facilities. The last three steps are peculiar to the system described but they are easier to carry out than their equivalent in most other systems which often need computer assistance to make them viable.

A method of calculating the pattern length and the repetition interval ($E_K$) (step (c)) above is illustrated in part of Fig 3. This involves the rounding up of the sum of the products of the relative sampling rates and the number of sources of each rate to an integer which is an integral multiple of the highest relative rate in use (see section 2.1). This number when divided by the highest relative rate gives the value of $E_K$ .

### 2.6 Advantages of the method

The method of controlling the order of sampling of data sources in a time shared system described above offers a number of advantages over previously used systems and other methods. This new approach gives the user considerable freedom of choice of sampling rates whilst providing extremely simple programming procedures. Only the minimum of program storage is needed and the user requirements are automatically converted into a viable sampling pattern which may be very long. The method can be implemented in a form which enables a high upper limit to the overall sampling rate to be achieved. This limit is virtually independent of the number of sampling rates provided.

## 3 SYSTEM CONTROL

### 3.1 Philosophy

The aims of the work described in this Report, as detailed in the introduction, were not fully defined at the outset but developed as it became clear what it was possible to achieve. The first priority was to develop a satisfactory multiplexing switch control system. Once a solution to this problem had been realised (see section 2) attempts were made to improve the overall procedure involved in the gathering and subsequent recovery of experimental data.

The first objective was to make the data record, probably on magnetic tape, self contained so that it could be demultiplexed with the minimum of reference to external records. The most practical way of achieving this end in the 085 prototype equipment was to arrange to record the addresses of the sources, in place of the data, on suitable occasions such as power switch on and when the sampling program was changed during a run. This process is termed a 'dump sequence' and is described in more detail in the next section. In use it was found to suffer from some operational difficulties which were removed as a result of the introduction of the new edit facility (see section 3.3).

The edit facility involved the separation of the synchornising, timing and housekeeping functions from that of data gathering which enabled a selective high speed data recovery process to be implemented. The resulting system satisfies the overall objectives of the development.

### 3.2 Dump sequence

The provision of the ability to change the sampling program during a trial (see section 2.4) inevitably led to the need to know, when recovering the data, which program was in use at any time. The method of conveying the program identity must provide some information that is not embedded in the multiplexed data stream since demultiplexing is not possible until the program is known. The first solution to be implemented was called the 'dump sequence' which is illustrated in Fig 5. The necessary non-multiplexed information is provided, in this method, by the form and content of the sequence.

In this system the necessary items of housekeeping information were treated as data sources and were sampled in the same way. The information consisted of synchronisation words, a field count, a sub-field count (to enable long fields to be subdivided) and a composite housekeeping word (known as documentary data). The latter was derived from the settings of a number of ten position thumb-wheel switches and a push-button on a control unit. One of these switches was used to select the required multiplexing program and the others were available for trial identification purposes. The push-button was used to enable a significant event to be marked by setting one of the bits to a logic one for a specified time interval.

The dump sequence was initiated when power was applied to the system and when the program selector switch was moved during a trial. Once initiated the entire dump sequence (see Fig 5) was transmitted or recorded in place of sampled data. The sequence commenced with unspecified information to the new field format which was continued for a fixed duration to allow time, during replay, for the relevant electronics to settle. This was followed by the main part of the sequence which consisted of an introductory section of seven fields of zeros and one of ones followed by eight fields in which the source addresses were used in place of the normal data. The synchronising words were sampled normally throughout the sequence to enable the data recovery system to function.

The demultiplexing process used the information contained in the address fields of the dump sequence to determine the identity of each data word. The system could therefore function without the need for external information except for a knowledge of the allocation of the system channels to the various data sources, and would automatically adjust the demultiplexing parameters each time a SCAN change was detected.

Although the introduction of the dump sequence represented a significant step forward and systems using it have performed a useful service, it suffers from three limitations which affect its operational usefulness. First, since all the housekeeping information is sampled in the same way as the data sources, it is not possible to demultiplex the data without first processing a dump sequence to find the location, within a field, of the housekeeping information and hence the identity of the program in use. Secondly, there is no absolute identification attached to the data. The field count, which incremented once per field, was used as a measure of elapsed time but, because it was reset during each dump sequence, the value could be ambiguous unless care was taken to count the number of preceding dump sequences. Under some operational conditions it could be difficult to count the number of dump sequences accurately and in any event it is not efficient operationally to start all replay runs from the beginning of the record. Thirdly, increases in system capacity and number of relative sampling rates available made very long fields possible which posed difficulties in both extracting a verified address field from the eight address fields carried by the dump sequence and the subsequent, software based, demultiplexing. These operational restrictions were removed by the introduction of the EDIT FACILITY which is described in the next section.

## 3.3 Edit facility

### 3.3.1 Format

To overcome the operational limitations of the dump sequence approach and to provide the high speed editing capability needed for use with high data capacity systems a modified data word format was introduced. This enabled the synchronising and housekeeping information to be separated from and conveyed independently of the multiplexed data. Because the easy access to the 085 housekeeping data during data recovery enabled the rapid location and extraction of specific sections of data this new approach was called the edit facility.

The original data word consisted of 14 binary digits comprising one parity, 12 data and one spare. When the data was recorded in parallel form on magnetic tape the spare bit occupied a track on its own. This was renamed the edit track and was used to carry serial 64 bit edit words (see Fig 6). The elements of this word are described in the next section. If the data were to be transmitted or recorded in serial form the 64 bits of an edit word would be distributed throughout the data. However, after serial word synchronisation and serial to parallel conversion, the edit word would again appear as a simple serial bit stream separate from the parallel multiplexed data words.

### 3.3.2 Information carried

Information carried by the edit word is, because of the known and fixed format, easy to extract during recovery. It should therefore carry information relating to the scan in use and some unique means of identifying the data so as to remove the disadvantages associated with the earlier dump sequence (described above). In addition, if the other housekeeping functions such as synchronisation and trial identification could be accommodated, the task of the user would be considerably simplified because he would then need to concern himself only with his own data sources.

The various elements that were chosen and their position within the edit word are shown in Fig 6. The synchronising word (SYNC) together with the edit word count provide the necessary synchronising function and are described in detail in the next section. The two groups of four bits, labelled ADDRESS and BCD, are used to convey the settings of up to sixteen thumb wheel switches mounted on a control unit. Each switch is identified by a number, its address, and it is this number together with the binary coded decimal value of the switch setting that is carried by the edit word. The switches are interrogated sequentially and it takes sixteen edit words to convey all the values. One switch is used to select the sampling program (scan), some are available for trial identification and the remainder are used to enter the time of start of the trial (start GMT). Elapsed time is provided by the 16 bit REAL TIME COUNT (RTC) which is incremented once per second and thus provides a duration of 18 hours. Start GMT and the value of RTC can be used to identify unique positions in the data record provided the correct operational procedure is carried out.

The event bit or flag, that was provided in the earlier systems (see section 3.2), has been replaced by an EVENT COUNT which greatly simplifies the location of a specific event. The flag bits are used to simplify the data recovery task.

### 3.3.3 Synchronisation

To enable demultiplexing to be carried out any multiplexed data stream must carry information that enables the start of a pattern to be identified. In addition it is desirable, in the case of long patterns, that demultiplexing is able to commence some way through a pattern without reference to the start thus ensuring that the minimum of data is lost after an interruption in the data flow. If data, such as the edit word, is transmitted as a serial bit stream the synchronising information must also enable the start of each word to be identified. The problem of serial word synchronisation will be discussed first followed by that of sampling pattern identification.

In the case of a serial bit stream it is normal practice[8] to insert a prearranged synchronising pattern into the stream at regular intervals. The ability of a system to first acquire and then maintain synchronism depends on the length and content of the synchronising pattern, the separation between patterns, the nature of the information carried by the serial stream, the characteristics of the data link and the method used for locating the pattern during data recovery. An example of a simple synchronising pattern detector is shown in Fig 7a. The serial data stream is passed through the shift register. Each shift causes the contents of the register to move one place to the right. After each shift the contents of the register are compared with the synchronising pattern and the number of pairs of similar bits counted to give the number of agreements. The variation of the number of agreements as the synchronising pattern passes through the detector is shown in Fig 7b. The number is dependent on the values taken by the data bits and may have any value in the shaded area. In the case illustrated in Fig 7a the five bits of the synchronising pattern that have entered the register contribute only two agreements whilst the two remaining data bits could add between 0 (DD = 00) and 2 (D = 11) more giving a total of between two and four agreements as indicated in Fig 7b. It can be seen that the number of agreements (seven here) that occur when the synchronising pattern is fully in the register is not unique and can occur in other positions depending on the values taken by the data bits. The information about the number of agreements must be supplemented by a knowledge of the repetition rate of the synchronising pattern if false synchronisation is to be avoided. If any of the synchronising bits in the data stream are complemented, due for example to noise in the data link, the number of agreements when the pattern is present will be reduced below seven whilst the number in adjacent bit positions will increase. The design of the Decision-Logic (Fig 7a) that generates a synchronising signal based on the number of agreements is usually a compromise between complexity and performance.

It is the possibility that any pattern or sequence of binary digits can occur in a serial stream of data that gives rise to the design problems and non-ideal performance of synchronising systems. In the case of the edit word most of the information carried by the serial stream is in the form of numbers which, although they may take any value, vary in an orderly manner. For example the values of the real time clock will either remain the same or increment by one between edit words. A synchronising scheme has been developed which takes advantage of this orderly behaviour and provides a characteristic, in which the synchronous condition is unique.

The new method resulted from an attempt to maximise the difference between the system's response to the synchronising pattern and to the data carried by the edit word. The fact that most of the edit word data only varied slowly and that significant portions remained the same in successive edit words suggested that the difference between the data and the synchronising pattern would be increased if the latter were to be changed between edit words. Accordingly the number of synchronising pattern bits that changed between edit words was maximised by using the ones complement of the pattern in place of the pattern itself in alternate edit words (*eg* 1110010 followed by 0001101).

A modified form of the synchronising pattern detector shown in Fig 7a was devised and is shown in Fig 8. The number of stages in the shift register was increased to enable two successive synchronising patterns to be examined together thus taking advantage of the difference between the steady data and changing patterns in successive edit words. The new detector consists of two of the simple detectors connected together via a digital delay line (B in Fig 8). The first detector compares the contents of section A of the shift register with the synchronising pattern whilst the second compares the contents of section C with the ones complement of the pattern. The outputs of the two detectors are
085 summed to give the number of agreements. The variation of this number as edit words, carrying fixed data, are passed through the detector is shown in Fig 9a. When section A of the shift register contains the synchronising pattern and C its ones complement there are 32 agreements. Alternately when section C contains

the pattern and A the complement there are no agreements. When sections A and C contain the same fixed data, bits in A that agree with the synchronising pattern must, in the same position in C, disagree with the complement and *vice versa*. Fixed data will therefore give rise to a constant 16 agreements. Whilst the synchronising patterns are entering and leaving sections A and C of the shift register the number of agreements will vary in a manner which is dependent on the actual pattern used.

Obviously the data carried by the edit word will vary and the number of agreements will depart from the value 16. For example if one bit changes between edit words it will contribute either two or zero agreements instead of the one that would have occurred if the bit had not changed. Three steps were taken to limit the effect changing data had on the synchronising characteristic. First a cyclic code was used for the edit word information in order to minimise the number of changes that occurred when the value of an edit word entry (*eg* the real time clock) increased by one. Secondly, the positions of the various items of information within the edit word were chosen to reduce the effect of the items that could change in a random manner (such as BCD, see Fig 6 and section 3.3.2) between edit words. Thirdly, an optimum synchronising pattern was derived, using a computer simulation, that ensured that the number of agreements, even with worst case data, would be between eight and 24 except when the synchronising patterns were present in A and C.

The resulting variation of the number of agreements is shown in Fig 9b. To simplify interpretation the characteristic has been folded up along the level of 16 agreements and then shifted down to the origin (*ie* y = |number of agreements - 16|) . It can be seen that when correct alignment occurs a level of 16 is achieved whereas in all other positions the number is eight or less. Thus there is a unique condition when synchronism exists which is tolerant of several bits in error.

It was pointed out at the start of this section that both sampling pattern and serial word synchronising information were required. Once an edit word has been isolated, with the aid of the synchronising word detector described above, the contents may be examined. One of the items is the edit word count (see Fig 6) which provides the necessary pattern synchronising information. This number is reset to zero in the first edit word in a sampling pattern and is incremented by one in each successive edit word (*ie* every 64 data words). In operation, once a serial synchronising pattern has been detected and the edit word count extracted, the slot numbers of each of the data words associated with that particular edit word may be determined. Thus demultiplexing may be achieved without reference to the start of the pattern.

### 3.4 Program store dump sequence

The introduction of the edit facility enabled a change to be made in the form of the dump sequence (see section 3.2). It was felt that the automatic and self-contained features of the old sequence should be retained but that the problems associated with program-dependent and possibly long pattern lengths should be avoided.

The method adopted was that of transmitting or recording the contents of the Scan and Translator Stores in place of the data using a fixed pattern format. This process was repeated for a duration long enough to ensure error free recovery. The fixed and relatively short (typically 4352 words) pattern format combined with the unique marker (Flag 1) considerably simplified the extraction and verification of the store contents. If, as was intended, this sequence is recorded at the start of a tape it will provide when replayed information on all the sampling programs that may occur later along the tape record. The identity of the particular program in use at any instant is obtained from the edit word.

## 4 DEMULTIPLEXING METHODS

### 4.1 Introduction

The source or identity of a given word in a time shared data stream must be established before use can be made of the information it carries. This process of identification is known as demultiplexing and may vary from the simple to the complex depending on the particular requirement and the facilities available.

The pattern length and number of sources involved directly affect the amount of information storage required and, if this is too large, the form of the demultiplexing procedure needed to reduce the storage requirement to an acceptable level. The speed, or word rate, required of the demultiplexer largely determines the choice between the use of a computer for demultiplexing (*ie* using software) and the use of special purpose hardware. The form and complexity of the software or hardware is to a large extent dependent on the versatility required.

)85 Two distinct approaches are described in the following sections, a simple software system that was used with the low capacity systems already described and a comprehensive method that is suitable, when implemented in hardware, for use with high capacity systems equipped with the edit facility (see sections 3.2 and 3.3).

## 4.2 A technique for low capacity systems

A simple demultiplexing process for the dump sequence type of system was implemented in software using a general purpose mini computer although some associated functions such as parity checking and synchronisation word detection were performed externally using special purpose hardware. Only the general approach will be considered here because the actual method used in this type of system is very dependent on the performance required (speed of operation, protection against and recovery from drop-outs in the data, etc) and the speed and capacity of the computer and its peripherals.

In the case of low data rate systems that are used for limited periods (*eg* 1000 samples per second for two hours) it is reasonable to transcribe all the data onto computer compatible tapes in which form it would be suitable for further analysis. In this type of application demultiplexing of the data would normally be required in order to provide 'quick-look' facilities in the form of trace records of selected channels via an ultra-violet recorder and digitally on numeric displays to permit decisions on what analysis should be done. Certain demultiplexed information (*eg* housekeeping) could also be used to index the computer compatible tapes. Obviously the user must specify his demultiplexing requirement prior to a replay pass. This specification would define which data channels should be routed to which display device and the information would be held in the form of a data array in the computer memory.

In the type of system under consideration the information required for demultiplexing purposes is carried by the dump sequence (see section 3.2) which immediately precedes the multiplexed data. Since the multiplexing program may be changed at any time a dump sequence could occur anywhere within the data record and a continuous check must therefore be made, during a replay pass, to determine whether a dump sequence is about to start. This involves searching for the characteristic data pattern of logic zeros followed by ones that occur at the start of a dump sequence (see Fig 5). When the onset of a dump sequence is detected, any demultiplexing activity based on an earlier dump sequence should be terminated, the system should be re-initialised and demultiplexing, based on the information carried by the sequence, should be commenced.

The initialisation of the system involves the extraction of that information, from the dump sequence, that is required by the demultiplexing process. This will normally involve deteimining the field length and the positions, within the field, of samples from channels that require to be demultiplexed. The field length may be determined by counting the number of words that occur between field synchronisation words. If several fields are examined the correct value can be obtained even if occasional errors occur. The field, in the dump sequence, containing all logic ones indicates, when detected, that in the next eight fields the sampled data is replaced by the channel addresses. The slot number of each word may be determined by counting the number of words that have occurred since the last field synchronising word. During each word period the channel address present may be compared with the list of channels required by the user and if needed the current slot number may be recorded in the computer memory. If a parity error occurs before the field is completed the process should be aborted and restarted at the beginning of the next field. Once an error free field is completed all the information needed will have been extracted but may need rearranging before demultiplexing can commence.

Demultiplexing may be accomplished in several ways. One simple method involes establishing the slot number of each data word by counting from the start of the field. Each slot number is compared with the first entry in a list of slot numbers that contain samples of interest. When equality occurs the data sample present in the slot is routed to the appropriate output channel. The next and subsequent slot numbers are compared with the second entry in the table until that slot is reached and the data outputted. This process continues to the end of the list and restarts when the next field starts.

If the acquisition system samples a large number of channels using a wide range of relative rates, the number of slots containing data samples of interest may be larger than the available memory capacity. In this case the slots of interest may be calculated from a knowledge of the first slot occupied by a given channel and the numbers of slots between successive samples. The number of words of storage required to implement this method is three times the number of channels to be demultiplexed and comprises, for each channel, words specifying the first slot occupied, the increment between occupied slots and a working register containing the value of the next slot occupied. At the start of a field the 'next slot' value equals the 'first slot' number and is increased by the 'increment' each time its current value is reached. This method obviously involves computation and either sorting or additional comparisons and will therefore take longer.

The system that used this approach has worked satisfactorily and at the desired speeds. However its success is due in the main to the low error rate of the system, of the order of one word in $10^5$. More sophisticated methods of achieving and maintaining synchronism could have been adopted but would have undoubtedly slowed the process down. Since all the data on the flight tape was transcribed to the computer tape occasional demultiplexing errors only affected the data block header information which was only intended as a guide to the user. It was at this point that the decision to adopt the edit facility was made and more sophisticated synchronising procedures were built into an edit word processor.

### 4.3 A technique for edit facility systems

The synchronisation information in the edit facility system is carried by the edit stream (see section 3.3.3) and theoretically if one complete edit word plus 16 bits (the length of a synchronisation word) of the following word are recovered without error it is possible to establish the slot number of each data word within this group of 80 words. In practice two or more edit words may be required in order to establish the validity of the value of the recovered edit word count which is used to establish the slot number, nevertheless synchronisation could be re-established rapidly after the reappearance of data following a fault condition.

To take advantage of this synchronisation characteristic a means of demultiplexing based on slot number was required. To enable high speed operation to be achieved and to ensure rapid recovery after a fault condition the demultiplexing process should not need to refer back to the start of the pattern and its speed should be independent of pattern length. A method has been developed which uses an iterative relationship to determine the identity of a data sample in a given slot.

The slot number S of a data sample, which has been derived from a knowledge of the edit word count (see above), is first divided by the repetition rate of samples from sources sampled at the highest rate ($E_K$ , see equation (3), section 2.1) to give Q , the integral part of the quotient, and a remainder R . Q is then expressed as a binary number thus;

$$Q = \sum_{\ell=1}^{K-1} 2^{K-1-\ell} q_\ell \qquad (4)$$

where the $q_\ell$ are binary digits. These digits are then used to calculate the values of a series of parameters $Z_\ell$ which are related by the expression;

$$Z_\ell = Z_{\ell+1} + q_\ell(\omega_{\ell+1} - N_\ell) + \bar{q}_\ell(- N_\ell) \text{ ,} \qquad (5)$$

where the suffix $\ell$ indicates a value corresponding to rate $\ell$ . The physical significance of the quantities in equation (5) can be explained by reference to Fig 10, which also illustrates how the equation is derived. The figure shows the sampling pattern divided into repeating sections at the kth sampling rate by the rate resets for that rate (see Fig 3). The corresponding digit $q_k$ has the value 1 for the slots in the second half of each section (*eg* $q_3$ is 1 for slots 3, 4, 5; 9, 10, 11; 15, 16, 17 and 21, 22, 23 in Fig 3). The relationships between $q_{k+1}$ and $q_k$ and their respective sampling rate sections of the pattern can be seen in Fig 10.

$Z_k$ is defined as the number of slots not used at rate k and above, in the section in which slot S falls (and may therefore be used at lower rates). If the slot S falls in the first half of the section at the kth rate ($q_k = 0$) the value of $Z_k$ is simply the value given by the next higher rate ($Z_{k+1}$) less the number of slots used at the kth rate (*ie* $N_k$). If the slot S falls in second half of the section ($q_k = 1$ , as is illustrated in Fig 10) the value of $Z_{k+1}$ is augmented by an amount $w_{k+1}$ which corresponds to the number of unused slots in a complete section at the (k + 1)th sampling rate. The value of $Z_k$ is therefore given by the relationship already quoted (see equation (5)). This is a simple iterative relationship which involves adding one of two constants, which may be negative, to the previous value. Which constant is used at each step depends on the value of the appropriate binary digit $q_\ell$ (since $\bar{q}_\ell = \text{NOT } q_\ell$) . The values of $Z_\ell$ are calculated for decreasing values of $\ell$ , starting with $\ell = K$ (the highest rate), until a negative or zero value is obtained. This condition indicates that the slot contains a sample taken at that particular value of $\ell$ and hence defines k . A value of zero corresponds to the last source to be sampled at this rate (*eg* SW7 of Fig 2b) and the more negative the value the earlier the source in the sequence (*eg* - 2 would indicate SW5 in this figure).

To calculate the values of $Z_\ell$ the initial value and the values of the constants are required. The initial value is $Z_K$ and is given by:

$$Z_K = R + 1 - N_K \qquad (6)$$

where R is the remainder of the division described above and $N_K$ is the number of sources sampled at the highest rate. The 1 is required because the slots are numbered starting with 0 (see Fig 3). The value of $w_K$ is given by

$$w_K = E_K - N_K \qquad (7)$$

and the other values may be calculated using the relationship;

$$w_\ell = 2w_{\ell+1} - N_\ell . \qquad (8)$$

If all the values of $E_\ell$ are found to be positive then the slot concerned must have been used as padding (*eg* slot 23 in Fig 3).

Once k is known, the identity of the sample in the form of the appropriate translator address T , can be calculated from the formula;

$$T = SP_k + Z_k \qquad (9)$$

where $SP_k$ is the stop number for the rate k .

If the required destinations of data from each data source are held in a Destination Store, in locations corresponding to those occupied by the source addresses in the Translator Store (see Fig 2), the address of the required destination may be obtained by adding the value of $Z_k$ to the address of the last entry in the block used at the kth rate. By this means a data sample may be despatched to several destinations simultaneously for example to a galvanometer channel and, via the computer, to a computer compatible tape unit. The actual word rate used during the demultiplexing process is normally limited by the peak data rate capabilities of the peripheral devices in use for the given run.

## 5 CONCLUSIONS

A versatile method of controlling the flow of data in time shared data acquisition and replay systems has been described. The multiplexing switch control system proposed employs a minimum of storage to generate sampling patterns automatically with little restriction on their length or complexity. The system is simple to program and imposes the minimum of constraints on the user.

A new technique which keeps the system control or housekeeping information separate from the multiplexed data has also been discussed. This considerably simplifies the user's task and yet enables the replay process to be optimised. This facility uses a novel synchronisation scheme which has unique properties.

A compatible high-speed demultiplexing scheme has been described which enables full advantage to be taken of the properties of both the multiplexing switch programmer and the control system.

Analysis of all the data gathered by a high capacity system is seldom required and economies can be readily effected by using a semi-automatic editing scheme to select the data required for processing. Equipment incorporating these concepts that requires little specialised knowledge to operate has been developed.

## LIST OF SYMBOLS

$E_k$ repetition rate of samples from a source sampled at the kth relative rate (*ie* samples from the source will occupy every $E_k$th slot)

$F$ field or pattern length

$K$ number of relative rates

$k$ the kth relative rate ($1 \leqslant k \leqslant K$)

$N$ total number of sources $= \sum_{k=1}^{K} N_k$

$N_k$ number of sources sampled at relative rate $r_k$

$P$ padding slots

$Q$ quotient of $S/E_K$

$q_\ell$ the binary digits of $Q_K$

$R$ remainder of $S/E_K$

$r_k$ the magnitude of the kth relative rate ($= 2^{k-1}$)

$S$ slot number $0 \leqslant S \leqslant F - 1$

$SP_k$ stop number for rate $k$

$T$ translator address

$w_k$ number of unused slots in a complete section at the kth sampling rate

$Z_k$ number of unused slots in the section at the kth sampling rate, in which $S$ falls

## REFERENCES

| No. | Author | Title, etc |
|---|---|---|
| 1 | - | General purpose flight test data acquisition system contracts K/LT216/443 and K/LT216/672 |
| 2 | - | Brochure describing the digital data acquisition systems for flight test manufactured by the Plessey Co Ltd |
| 3 | G.A. Cooper<br>K.D. Goddard<br>G.C. Lowe | Data signal switching apparatus.<br>United Kingdom Patent 1447241 |
| 4 | G.A. Cooper<br>K.D. Goddard | New techniques for controlling data flow during both the record and replay phases of airborne data acquisition.<br>In *Proceedings of the 8th International Aerospace Symposium Cranfield* 1975 |
| 5 | T.G.M. Dale | Newsletters Nos. 20 and 21.<br>Trials Department, RAE. August and December 1960 |
| 6 | G.C. Lowe | Low-level high-speed multiplexing switches for data acquisition systems.<br>RAE Technical Report 64033 (1964) |
| 7 | - | Common user instrumentation service telemetry standards.<br>Document TS26/74 (1974) published by EMI Electronics Ltd for DGWT MOD(PE) |
| 8 | R.H. Barker | Group synchronising of binary digital systems.<br>In communication theory, edited by W. Jackson, pp 273-287<br>Academic Press Inc, New York (1953) |

Documents quoted are not necessarily available to members of the public or to commercial organisations.

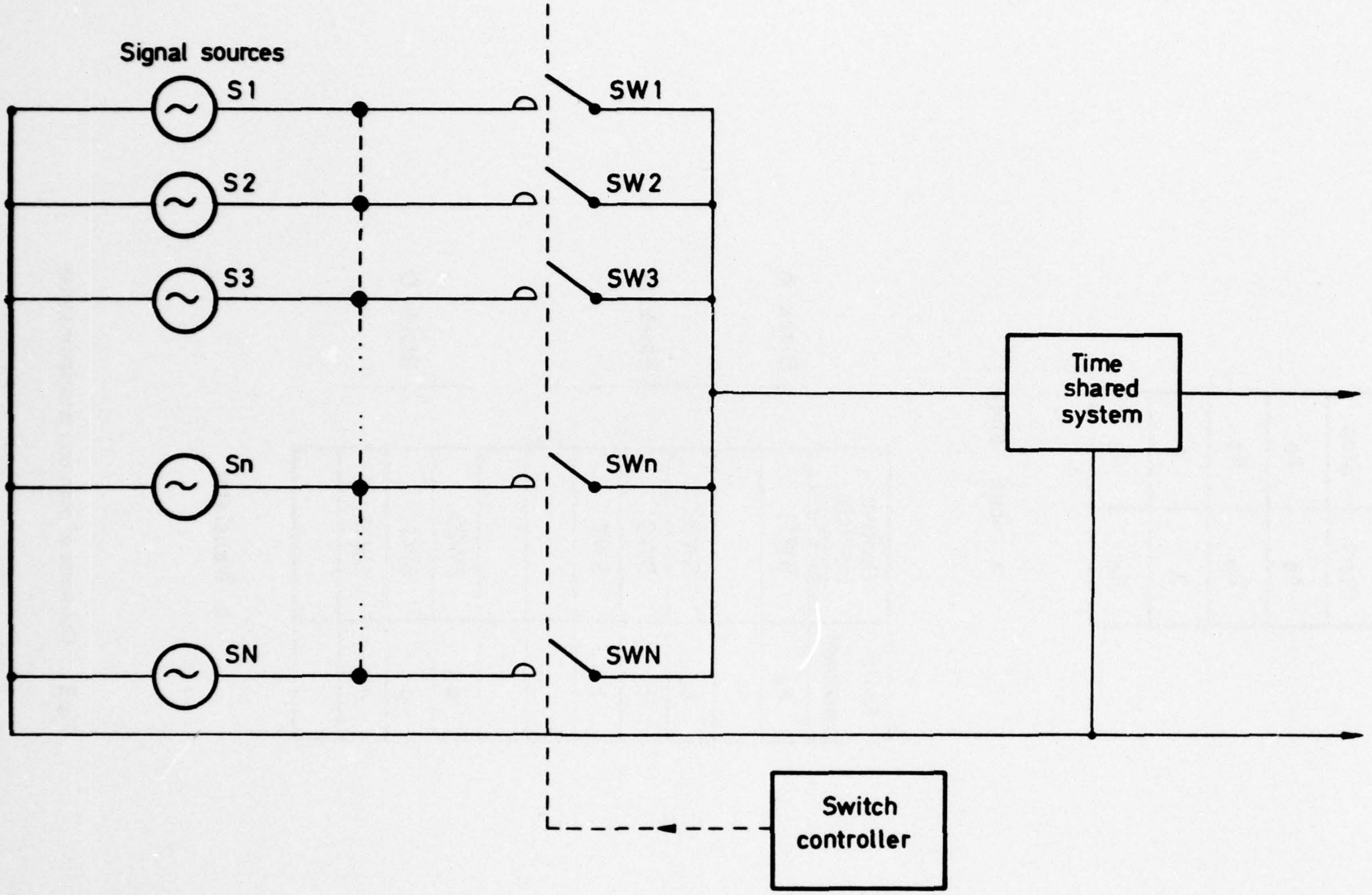


Fig 1 Schematic of a typical system

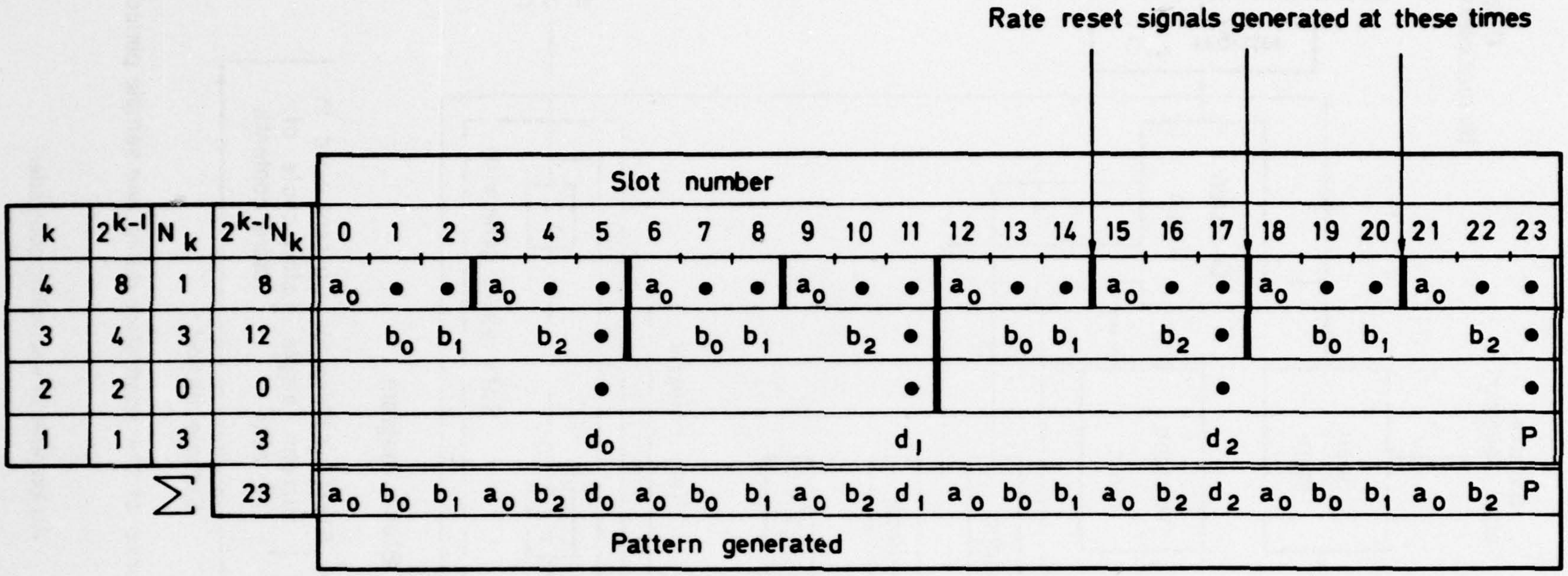
| Start | Stop |
|---|---|
| $a_0$ | $a_0$ |
| $b_0$ | $b_2$ |
| X | X |
| $d_0$ | $d_2$ |

a Scan store

| Store address | Content (switch address) | |
|---|---|---|
| $a_0$ | SW11 | Block A |
| | | |
| $b_0$ | SW5 | Block B |
| $b_1$ | SW2 | |
| $b_2$ | SW7 | |
| | | |
| | | |
| $d_0$ | SW20 | Block D |
| $d_1$ | SW1 | |
| $d_2$ | SW17 | |
| | | |

b Translator

Fig 2 Contents of scan and translator stores

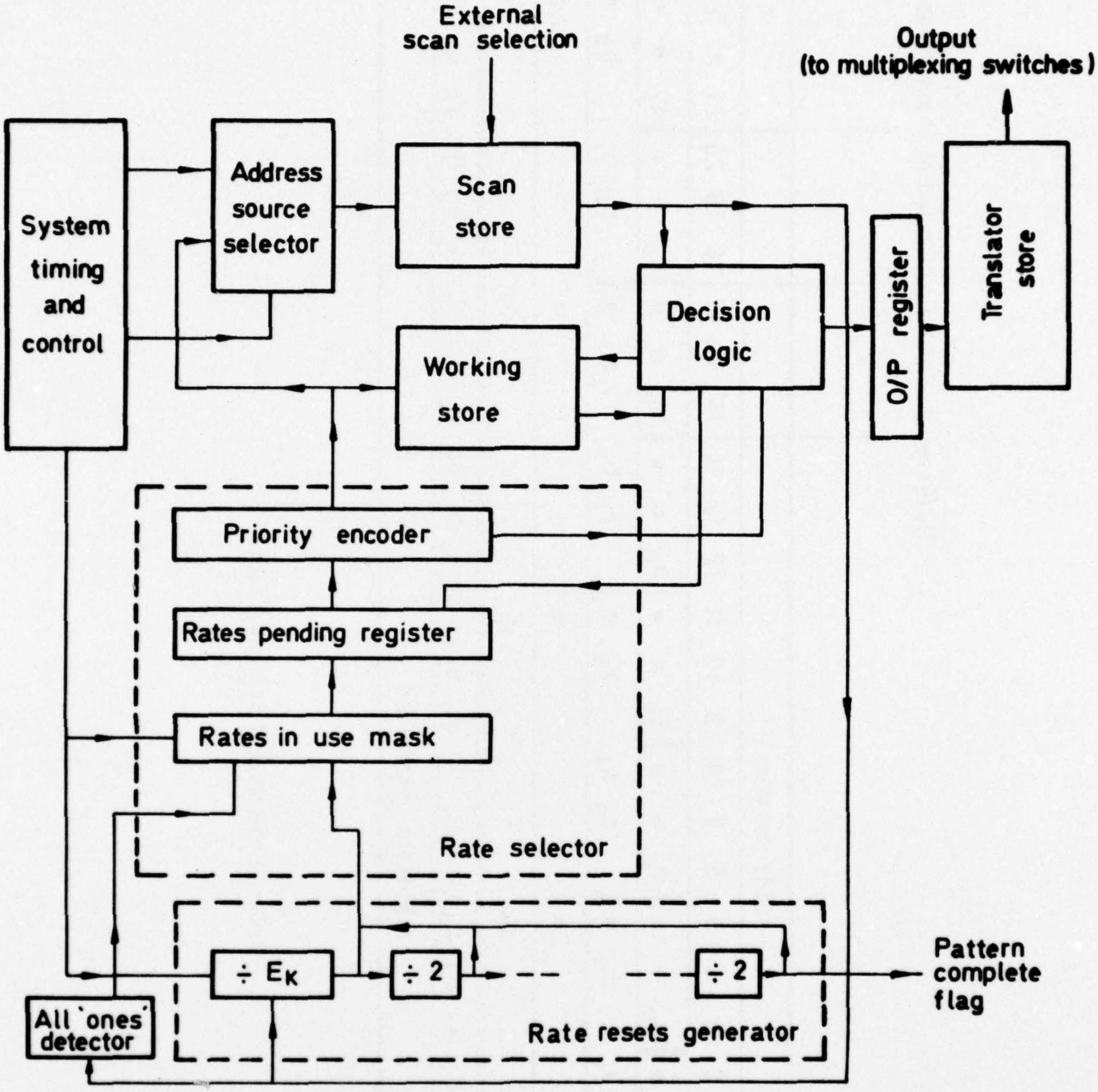


Rate reset signals generated at these times

| | | | | Slot number | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k | $2^{k-1}$ | $N_k$ | $2^{k-1}N_k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 4 | 8 | 1 | 8 | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● | $a_0$ | ● | ● |
| 3 | 4 | 3 | 12 | | $b_0$ | $b_1$ | | $b_2$ | ● | | $b_0$ | $b_1$ | | $b_2$ | ● | | $b_0$ | $b_1$ | | $b_2$ | ● | | $b_0$ | $b_1$ | | $b_2$ | ● |
| 2 | 2 | 0 | 0 | | | | | | ● | | | | | | ● | | | | | | ● | | | | | | ● |
| 1 | 1 | 3 | 3 | | | | | | $d_0$ | | | | | | $d_1$ | | | | | | $d_2$ | | | | | | P |
| | | $\sum$ | 23 | $a_0$ | $b_0$ | $b_1$ | $a_0$ | $b_2$ | $d_0$ | $a_0$ | $b_0$ | $b_1$ | $a_0$ | $b_2$ | $d_1$ | $a_0$ | $b_0$ | $b_1$ | $a_0$ | $b_2$ | $d_2$ | $a_0$ | $b_0$ | $b_1$ | $a_0$ | $b_2$ | P |
| | | | | Pattern generated | | | | | | | | | | | | | | | | | | | | | | | |

Fig 3 Generation of a sampling pattern

Fig 4



a Block diagram

| Stop address for current sample rate | Start address for current sample rate | Current word in slow cycle of store contents |
|---|---|---|
| Sample period | | |

b Output sequence of the scan store during one sample period

Fig 4 An example of a sampling controller

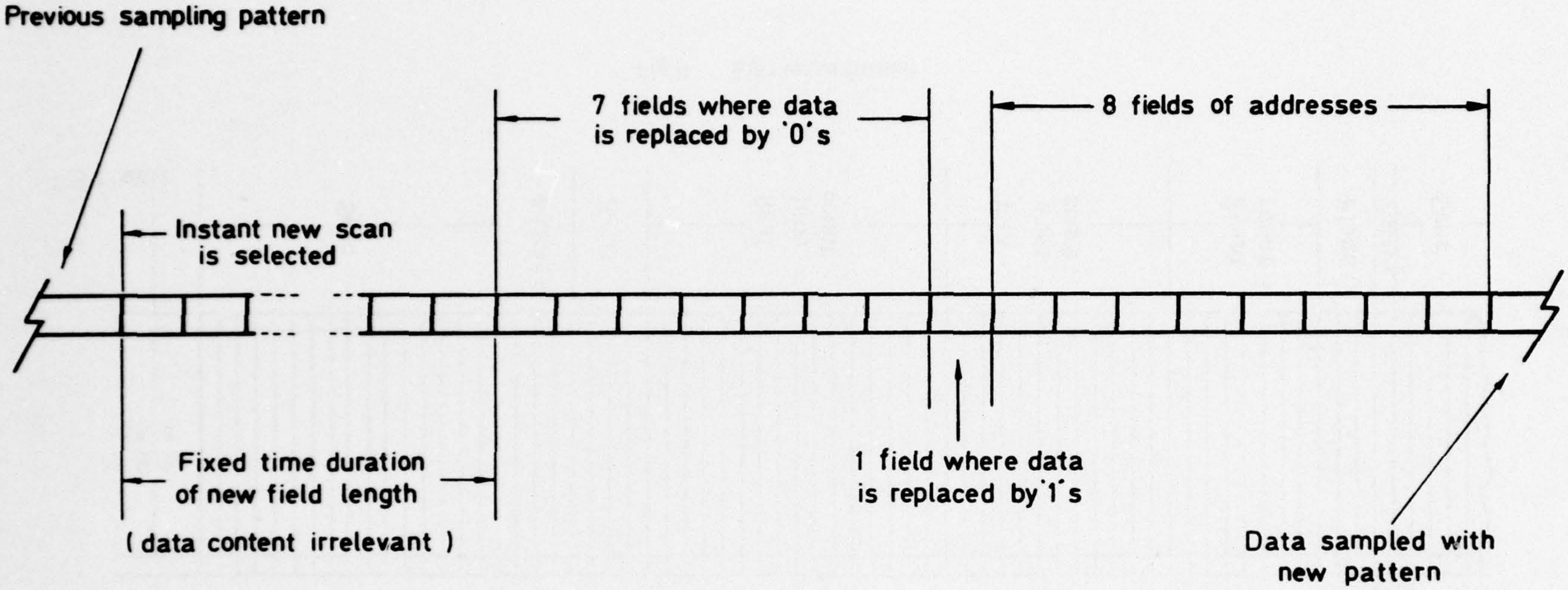

Previous sampling pattern
Instant new scan is selected
7 fields where data is replaced by '0's
8 fields of addresses
Fixed time duration of new field length
(data content irrelevant)
1 field where data is replaced by '1's
Data sampled with new pattern

Fig 5 Form of dump sequence

Fig 6

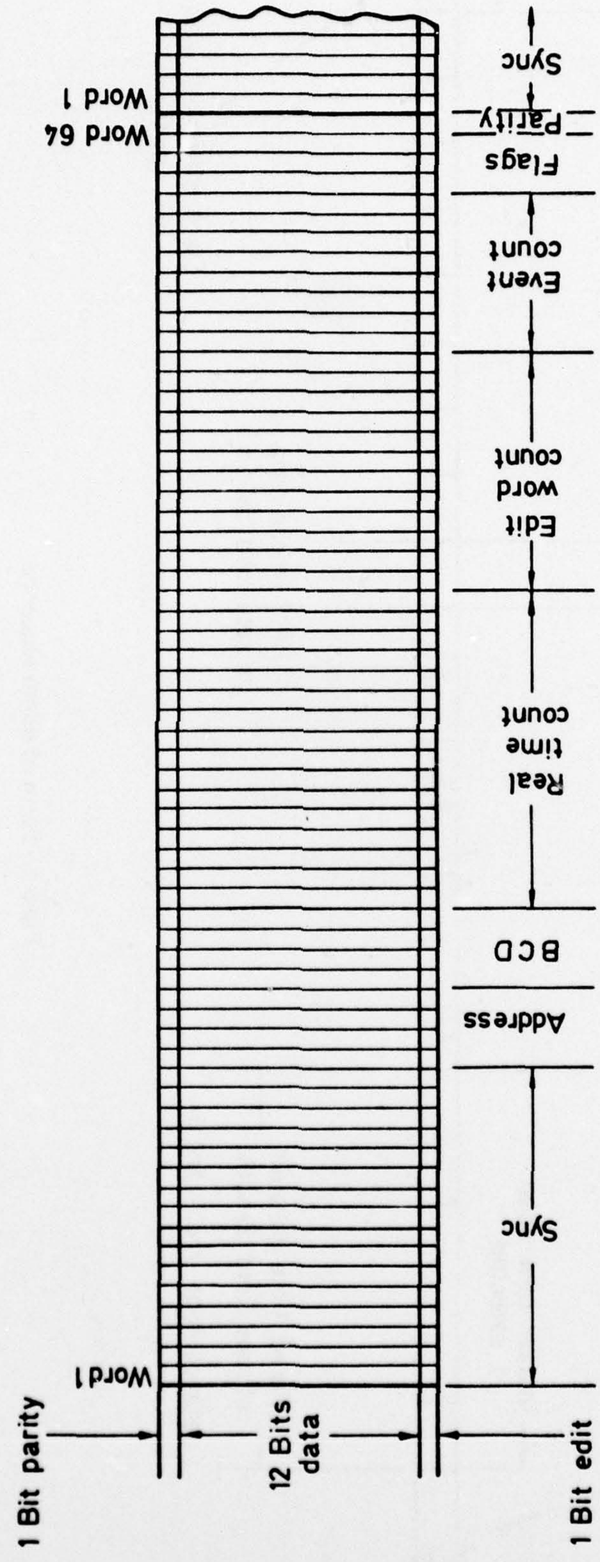
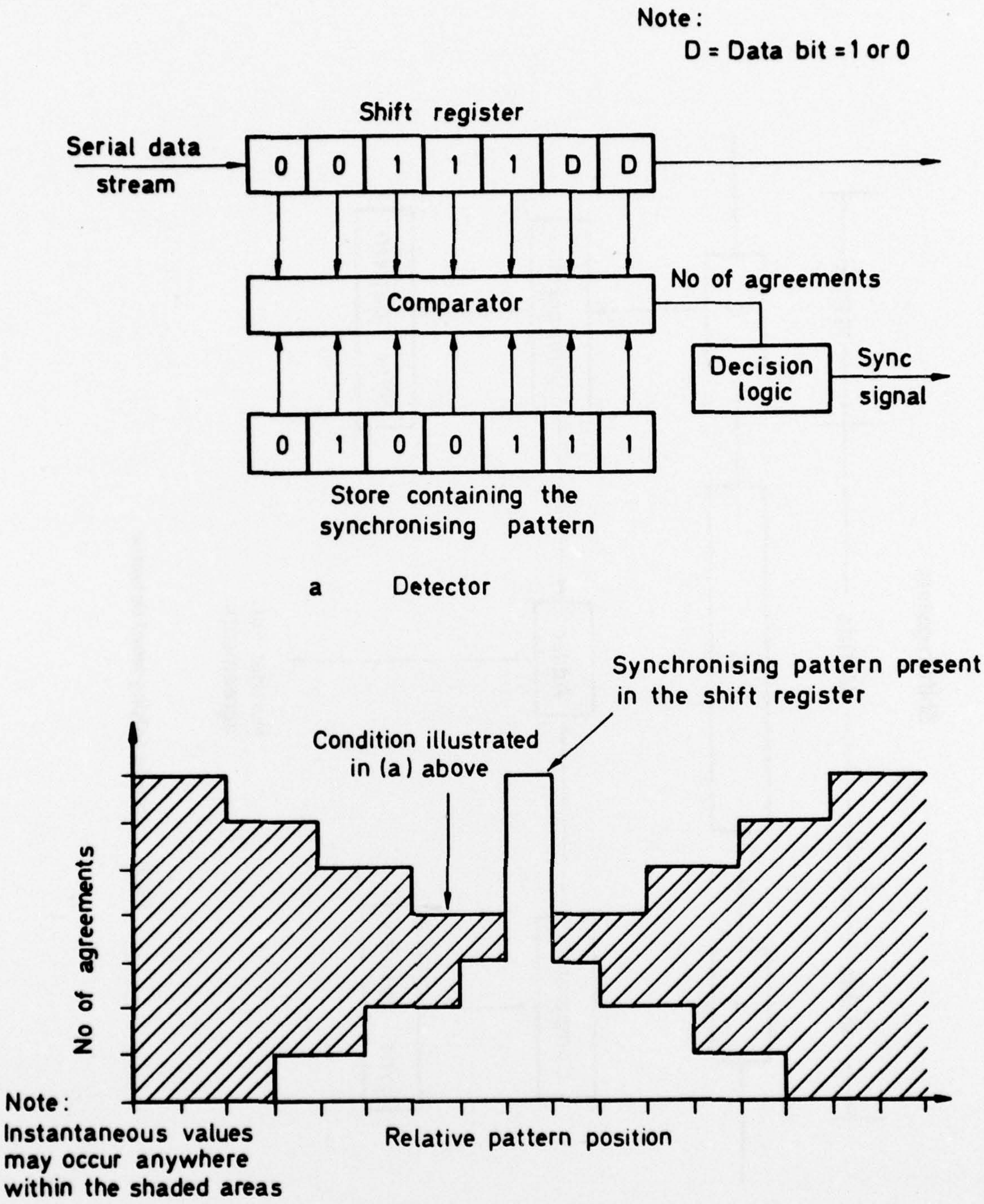


Fig 6 Edit word format


Note:
D = Data bit = 1 or 0
Shift register
Serial data stream
0
0
1
1
1
D
D
Comparator
No of agreements
Decision logic
Sync signal
0
1
0
0
1
1
1
Store containing the synchronising pattern
Synchronising pattern present in the shift register
Condition illustrated in (a) above
No of agreements
Relative pattern position
Note:
Instantaneous values may occur anywhere within the shaded areas

a Detector

b Variation of the number of agreements as the sync pattern is shifted through the detector

Fig 7 A simple synchronising pattern detector

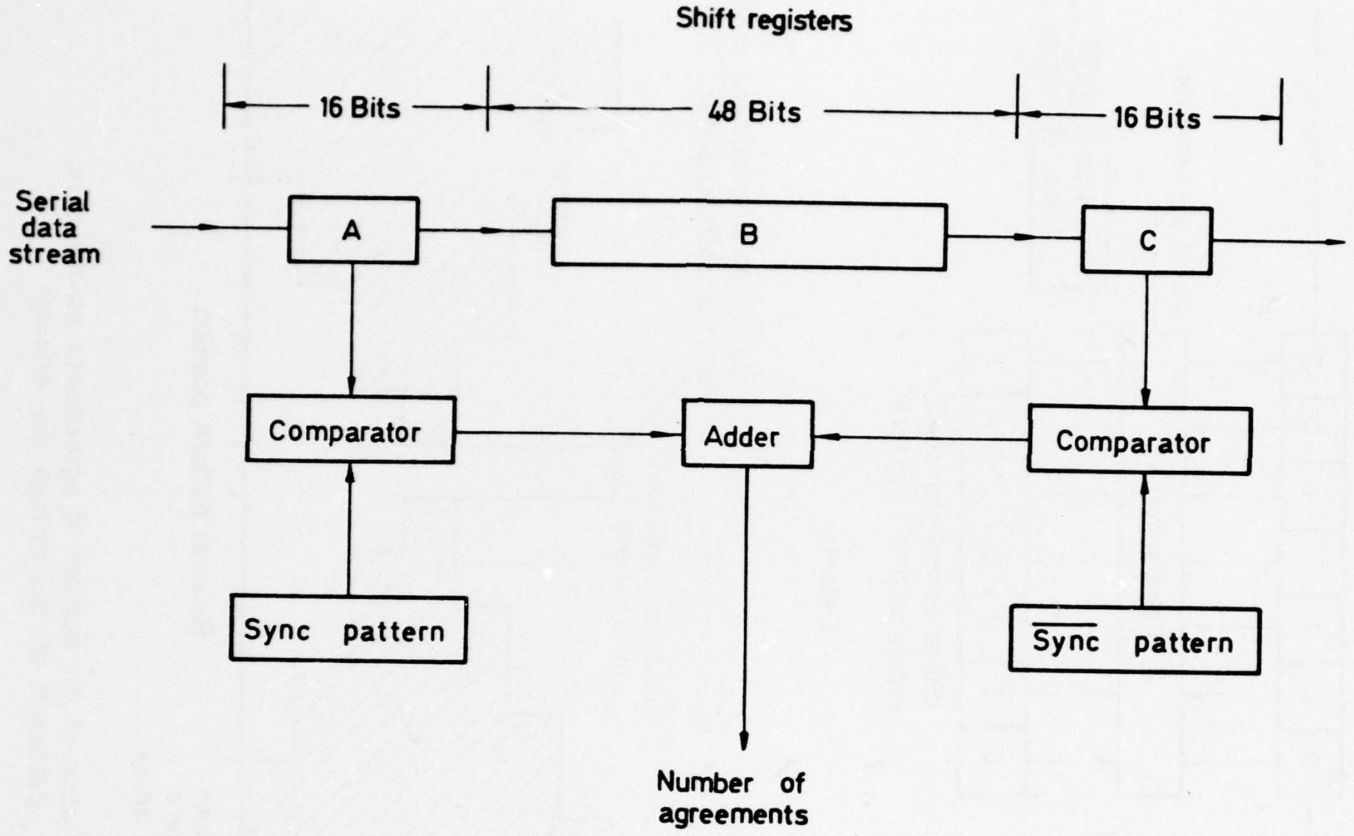
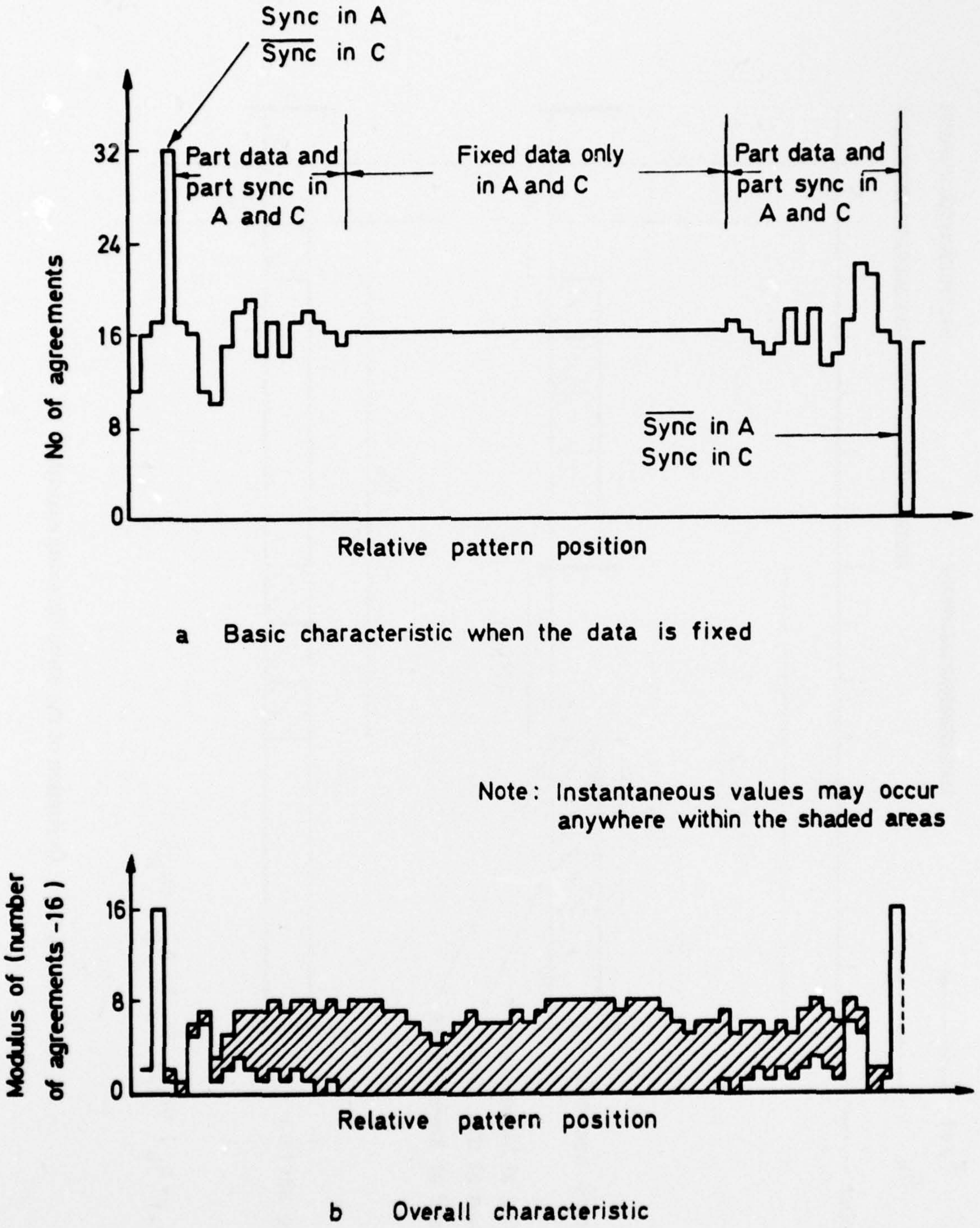


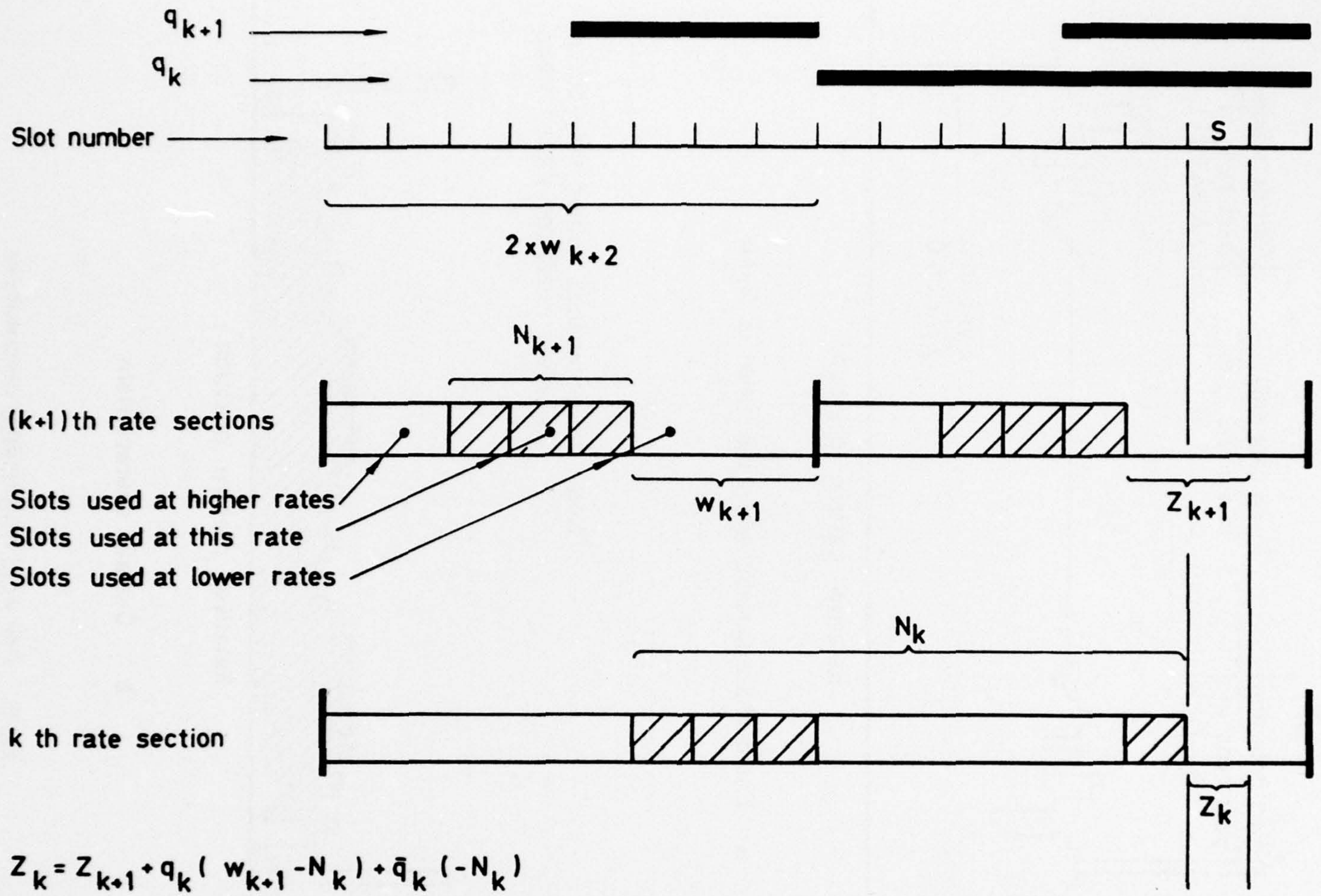
Fig 8 Synchronising word detector



a Basic characteristic when the data is fixed

b Overall characteristic

Fig 9 Edit word synchronising characteristic



$$Z_k = Z_{k+1} + q_k ( w_{k+1} - N_k ) + \bar{q}_k ( - N_k )$$

Fig 10 **Derivation of the demultiplexing algorithm**